AMENDMENTS TO THE CLAIMS

1. (currently amended) A virtual disk formatting system comprising:

a number of mass-storage devices, each having physical sectors of a first sector length; and

a routing component that provides a virtual disk interface to the mass-storage ~~components~~ devices by mapping each access operation ~~operations~~, received from an external entity ~~entities~~, each access operation directed to a virtual disk having virtual sectors of a second sector length, to one of the number of mass-storage devices having physical sectors of the first sector length.

2. (original) The virtual disk formatting system of claim 1 wherein the routing component is an integrated-circuit storage-shelf router.

3. (original) The virtual disk formatting system of claim 2 wherein the storage-shelf router provides a fibre-channel-disk-based virtual disk formatting interface to external entities and maps fibre-channel-disk-based access operations to a number of ATA disk drives included in a storage shelf containing the storage-shelf router.

4. (previously presented) The virtual disk formatting system of claim 1 wherein the routing component includes a processor and firmware/software programs that carry out virtual disk formatting.

5. (original) The virtual disk formatting system of claim 1 wherein virtual sectors are mapped onto contiguous physical sectors, allowing the physical sector and byte address of the first byte of a virtual sector to be calculated, when the second sector length is greater than the first sector length, as:

fsl = first sector length

ssl = second sector length

modulus = (smallest number evenly divisible by both fsl and ssl) / ssl

difference = ssl - fsl

$$\text{physical sector} = \text{virtual sector} + \left(\frac{\text{virtual sector}}{\text{modulus}}\right)$$

$$\text{physical byte address} = \text{remainder}\left(\frac{\text{virtual sector}}{\text{modulus}}\right) \times \text{difference}$$

and, when the second sector length is less than the first sector the physical sector and byte address of the first byte of a virtual sector is calculated, as:

fsl = first sector length

ssl = second sector length

modulus = (smallest number evenly divisible by both fsl and ssl) / ssl

difference = fsl − ssl

$$\text{physical sector} = \text{virtual sector} - \left(\frac{\text{virtual sector}}{\text{modulus}}\right)$$

$$\text{physical byte address} = \text{remainder}\left(\text{fsl} - \text{remainder}\left(\frac{\text{virtual sector}}{\text{modulus}}\right) \times \text{difference}\right) / \text{fsl}.$$

6. (original) The virtual disk formatting system of claim 5 wherein, when the modulus and difference are both evenly divided by 2, the division and multiplication operations can be replaced with shift operations, and the remainder operation can be replaced by a bit-wise and operation.

7. (original) A virtual disk formatting system comprising:

a number of mass-storage devices having physical sectors of a first sector length; and

a routing component that provides to external entities a first virtual disk interface to the mass-storage components by mapping access operations, received from the external entities, directed to the first virtual disk interface having virtual sectors of a second sector length to an internal, virtual disk interface with internal-virtual-disk-sectors having a third sector length larger than the second sector length,

and then mapping the access operations from the internal, virtual disk interface to the number of mass-storage devices.

8. (original) A virtual disk formatting system of claim 7 further including:

including, by the routing component, error detection information within the internal-virtual-disk-interface sectors in order to provide routing-component-mediated error checking.

9. (original) The virtual disk formatting system of claim 8 wherein the error detection information is a longitudinal redundancy check code.

10. (original) The virtual disk formatting system of claim 7 wherein the routing component is an integrated-circuit storage-shelf router.

11. (original) The virtual disk formatting system of claim 10 wherein the storage-shelf router provides a fibre-channel-disk-based virtual disk formatting interface to external processing entities and maps fibre-channel-disk-based access operations to a number of ATA disk drives included in a storage shelf containing the storage-shelf router.

12. (original) The virtual routing system of claim 7 wherein the routing component includes a processor and firmware/software programs that carry out virtual disk formatting.

13. (currently amended) A method for providing a virtual-disk-format interface to processing entities external to a number of mass storage devices, each having physical sectors of a first sector length, the method comprising:

providing a routing component; and

mapping each access operation ~~operations~~, received from an external entity ~~entities~~, each access operation directed to a virtual disk having virtual sectors of a second sector length by the routing component to one of the number of mass-storage devices having physical sectors of the first sector length.

14. (original) The method of claim 13 wherein the routing component is an integrated-circuit storage-shelf router.

15. (original) The method of claim 14 wherein the storage-shelf router provides a fibre-channel-disk-based virtual disk formatting interface to external processing entities and further including:

mapping, by the storage-shelf router, fibre-channel-disk-based access operations to a number of ATA disk drives included in a storage shelf containing the storage-shelf router.

16. (original) The method of claim 13 wherein the routing component includes a processor and firmware/software programs that carry out virtual disk formatting.

17. (original) The method of claim 13 further including:

mapping, by the routing component, virtual sectors onto contiguous physical sectors, allowing the physical sector and byte address of the first byte of a virtual sector to be calculated, when the second sector length is greater than the first sector length, as:

fsl = first sector length

ssl = second sector length

modulus = (smallest number evenly divisible by both fsl and ssl) / ssl

difference = ssl - fsl

$$\text{physical sector} = \text{virtual sector} + \left(\frac{\text{virtual sector}}{\text{modulus}}\right)$$

$$\text{physical byte address} = \text{remainder}\left(\frac{\text{virtual sector}}{\text{modulus}}\right) \times \text{difference}$$

and, when the second sector length is less than the first sector the physical sector and byte address of the first byte of a virtual sector is calculated, as:

fsl = first sector length

ssl = second sector length

modulus = (smallest number evenly divisible by both fsl and ssl) / ssl

difference = fsl − ssl

$$\text{physical sector} = \text{virtual sector} - \left(\frac{\text{virtual sector}}{\text{modulus}}\right)$$

$$\text{physical byte address} = \text{remainder}\left(\text{fsl} - \text{remainder}\left(\frac{\text{virtual sector}}{\text{modulus}}\right) \times \text{difference}\right) / \text{fsl}.$$

18. (original) The method of claim 17 wherein, when the modulus and difference are both evenly divided by 2, the division and multiplication operations can be replaced with shift operations, and the remainder operation can be replaced by a bit-wise and operation.

19. (original) A method for including additional information in disk sectors of a number of mass-storage devices having a first sector length, the method comprising :

providing a routing component;

mapping, by the routing component, access operations, received from external entities, directed to a first virtual disk interface having virtual sectors of a second sector length to an internal, virtual disk interface with internal-virtual-disk-sectors having a third sector length larger than the second sector length, and then mapping, by the routing component, the access operations from the internal, virtual disk interface to the number of mass-storage devices.

20. (original) The method of claim 19 further including:

including, by the routing component, within the internal-virtual-disk-interface sectors one of:

error-detection information;

additional information that, together with the data contained in the internal-virtual-disk-interface sectors, provides an encrypted version of the data directed to the first virtual disk interface by external processing entities; and

error-detection and error-correction information.